Multivariate Gaussian Process Emulators with Nonseparable Covariance Structures

Thomas E. Fricker

College of Engineering, Mathematics and Physical Sciences

University of Exeter

Exeter EX4 4QF, UK

(t.fricker@exeter.ac.uk)

Jeremy E. Oakley

Department of Probability and Statistics

University of Sheffield

Sheffield S3 7RH, UK

(j.oakley@shef.ac.uk)

Nathan M. Urban

Woodrow Wilson School

Princeton University

Princeton, NJ 08544, USA

(nurban@princeton.edu)

December 17, 2010

Thomas E. Fricker (corresponding author) is a Research Associate, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Harrison Building, North Park Road, Exeter EX4 4QF, UK (email: t.fricker@exeter.ac.uk); Jeremy E. Oakley is a Senior Lecturer, Department of Probability and Statistics, University of Sheffield, Sheffield S3 7RH, UK (email: j.oakley@shef.ac.uk); and Nathan M. Urban is a Research Associate, Woodrow Wilson School, Princeton University, Princeton, NJ 08544, USA (email:

Abstract

Gaussian process regression models or 'emulators' have become popular in the statistical analysis of deterministic computer models (simulators), in particular for computationally expensive models where the emulator is used as a fast surrogate. For models with multivariate output, common practice is to specify a separable covariance structure for the Gaussian process. Though computationally convenient, this can be too restrictive, leading to poor performance of the emulator, particularly when the different simulator outputs represent different physical quantities. Also, treating the simulator outputs as independent can lead to inappropriate representations of joint uncertainty. We develop nonseparable covariance structures for Gaussian process emulators, based on the linear model of coregionalization, and convolution methods. Using two case studies, we compare the performance of these covariances structures both with standard separable covariance structures, and with emulators that assume independence between the outputs. In each case study we find that only emulators with nonseparable covariances structures have sufficient flexibility to give both good predictions and represent joint uncertainty about the simulator outputs appropriately.

KEYWORDS: Computer experiment; Gaussian process; metamodel; convolved process; coregionalization.

1 Introduction

Deterministic computer models, or simulators, are used in many fields of science and technology to simulate complex processes. Statistical problems with using simulators include: assessing uncertainty in simulator outputs due to uncertainty in simulator inputs (McKay et al., 1979); predicting simulator outputs at untried input values for computationally expensive simulators (Sacks et al., 1989); identifying the 'most important' inputs using sensitivity analysis (Saltelli and Tarantola, 2002; Oakley and O'Hagan, 2004); calibrating simulators to noisy physical data (Kennedy and O'Hagan, 2001; Goldstein and Rougier, 2006). Santner

princeton.edu). This research is part of the "Managing Uncertainty in Complex Models" (MUCM) project, funded by Research Councils UK under its Basic Technology programme. The authors thank the other members of the MUCM team for helpful comments, Dr Neil Sims for providing the FEM data used in Case Study 1, and Dan Ricciuto for useful discussion about GPP modelling in Case Study 2.

et al. (2003) review these methods, and some recent developments are given in (Bayarri et al., 2009a).

For computationally expensive simulators, an important step is to build an emulator: a statistical model of the simulator's input-output relationship. The emulator is used as a surrogate for the simulator in solving the above statistical problems, so choices about the way the emulator is built have a critical impact on the subsequent analyses. It is particularly important that the emulator correctly represents the uncertainty in its predictions, so that the uncertainty about the simulator itself is properly accounted for and the user can establish whether further simulator runs would change any predictions or inferences. Regression methods are often used to construct emulators, with Gaussian process (GP) methods being a popular choice. Within the field of computer experiments, Sacks et al. (1989) first used GP regression for modeling simulator output, and Currin et al. (1991) gave a Bayesian treatment. Earlier, GPs were used in geostatistics for spatial interpolation in the technique known as *kriging* (Journel and Huijbregts, 1978). GPs have since become popular in the machine learning community for regression and classification (Rasmussen and Williams, 2006).

Early work on emulators was for simulators with one scalar output (*univariate emulators*), but attention has since turned to emulating simulators with more than one output. A simple approach is to consider the outputs as a collection of separate univariate functions, and emulate each function independently. It is likely, though, that multiple outputs from a single simulator will be related, so modeling prior beliefs about them as being independent may result in losing information. To capture this information, a *multivariate emulator* models the outputs jointly.

We categorize multivariate output from simulators into two classes. The first class comprises output from a simulator that simulates a quantity that varies over some continuous field, often space or time. Here, each output gives the value of the quantity at a particular location in the field. The index of the output specifies the location, and we can define a metric on the output index, for example spatial distance or time lag. We refer to this class

of multivariate output as *field output*. Kennedy and O'Hagan (2001) emulate field output by considering the output index as a new input to the simulator and using a univariate emulator with a stationary parametric covariance function. Conti and O'Hagan (2010), McFarland et al. (2008) and Bayarri et al. (2009b) use this approach. Rougier (2008) emulates field output directly but includes parametric regression terms on the output index, using the between-output metric. Higdon et al. (2008) calibrate a simulator with highly multivariate output, using principal components to reduce the dimensionality. Bayarri et al. (2007) consider simulators whose output is functional, which is a limiting case of field output. They use a wavelet representation of the function and emulate the wavelet coefficients with independent univariate emulators.

The second class of multivariate output arises from simulators that simulate different types of quantities simultaneously, and the index of the output is merely a label. We refer to this class of multivariate output as *multiple-type output*. A simulator with multiple-type output, considered in this paper, is the Simple Climate Model (SCM) (Urban and Keller, 2010). The SCM has three outputs: atmospheric CO_2 concentration, ocean heat uptake, and global surface temperature. Multiple-type output does not have an obvious measure of distance between the outputs, so it is unwise to model the between-output dependencies using a parametric covariance function. Traditionally, the outputs are either emulated independently or jointly with a separable covariance structure (e.g. Qian et al., 2008).

1.1 Outline and scope

This paper considers emulating multiple-type output. We focus on careful modeling of the between-output dependencies through the emulator covariance function. In section 2 we review the separable covariance approach. This appeals due to its mathematical tractability, but may be too restrictive when the outputs have differing correlation lengths. Our contribution is to propose two nonseparable covariance structures for multivariate emulators: one using convolution methods, and another using the linear model of coregionalization. In both cases we suggest adaptations of the standard implementation in order to make the

methods better suited for emulation.

In section 3 we present case studies, from engineering (using a finite element model of an aircraft), and from climate science (using a climate simulation model). We implement separable and nonseparable multivariate emulators, comparing them with each other and with multiple independent univariate emulators. We find that the separable multivariate emulator often results in poorer marginal predictions of single outputs than the independent univariate emulators. The nonseparable emulators retain the marginal advantages of the independent emulators while offering a multivariate treatment. The importance of the multivariate treatment becomes clear when we consider combining predictions of multiple outputs. In both case studies we consider a function of the outputs, demonstrating that ignoring the between-output dependencies leads to poor predictions of the function.

2 Approaches to multivariate emulation

Consider a deterministic simulator with a p-dimensional input $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$ and a vector of outputs $\mathbf{y} \in \mathbb{R}^k$, represented by the function $\boldsymbol{\eta} : \mathcal{X} \to \mathbb{R}^k$. Its outputs are *regular*, meaning that the same k outputs are observed every time the simulator is run. If $k = 1$, then the conventional approach is to represent the uncertainty in $\boldsymbol{\eta}(.)$ using a GP prior with a parametric covariance function. Kennedy and O'Hagan (2001) extend this model for $k > 1$ by including the output index as an additional input and proceeding with a univariate GP prior on the augmented input space $\mathcal{X}^* \subset \mathbb{R}^{p+1}$. The between-output covariances is dealt with by the extended covariance function $\mathbf{C}(.,.) : \mathcal{X}^* \times \mathcal{X}^* \to \mathbb{R}$. However, to use a parametric covariance function we need a metric for measuring the distance between two outputs. Hence this approach is most suitable for field output.

In this paper we are interested in multiple-type outputs. We assume that the outputs represent a variety of different quantities, so it is not sensible to impose a parametric form for the between-output covariances. Instead, we model the between-output dependencies

directly using a multivariate GP prior:

$$\begin{aligned} \boldsymbol{\eta}(.) &= \mathbf{m}(.) + \mathbf{z}(.), \\ \mathbf{m}(.) &= (\mathbf{I}_k \otimes \mathbf{h}(.)^T)\boldsymbol{\beta}, \\ \mathbf{z}(.)|\boldsymbol{\theta} &\sim GP_k[0, \mathbf{C}(.,.)]. \end{aligned} \tag{1}$$

In $\mathbf{m}(.)$, the prior mean function, $\mathbf{h}(.)$ is a vector of q regressors on $\mathcal{X}$ and $\boldsymbol{\beta}$ is a vector of kq unknown coefficients. The residual process, $\mathbf{z}(.)$, has a $k \times k$ matrix-valued covariance function, $\mathbf{C}(.,.)$, which is controlled by some hyperparameters $\boldsymbol{\theta}$.

The simulator is run at training design points $X = (\mathbf{x_1}, ..., \mathbf{x_n})$, giving training data outputs $\mathbf{y}^T = (\mathbf{y_1}^T, ..., \mathbf{y_k}^T)$, where $\mathbf{y_j}^T = (\eta_j(\mathbf{x_1}), ..., \eta_j(\mathbf{x_n}))$ is the vector of data from the jth output. The design is usually selected using some space-filling criterion; in our examples we use maximin Latin hypercube designs (McKay et al., 1979, Johnson et al., 1990). Conditioning on $\mathbf{y}$ and integrating over $\boldsymbol{\beta}$, we obtain

$$\boldsymbol{\eta}(.)|\mathbf{y}, \boldsymbol{\theta} \sim GP_k\{\mathbf{m}^\ddagger(.), \mathbf{C}^\ddagger(.,.)\}, \tag{2}$$

where, for a set of $\acute{n}$ new input points $\acute{X} = (\acute{\mathbf{x}}_\mathbf{1}, ..., \acute{\mathbf{x}}_\mathbf{n})$,

$$\begin{aligned} \mathbf{m}^\ddagger(\acute{X}) &= \acute{\mathbf{H}}(\acute{X})\hat{\boldsymbol{\beta}} + \mathbf{F}(\acute{X})\mathbf{V}^{-1}(\mathbf{y} - \mathbf{H}\hat{\boldsymbol{\beta}}), \\ \mathbf{C}^\ddagger(\acute{X}, \acute{X}) &= \mathbf{C}(\acute{X}, \acute{X}) - \mathbf{F}(\acute{X})\mathbf{V}^{-1}\mathbf{F}(\acute{X})^T + \\ &\quad (\acute{\mathbf{H}}(\acute{X}) - \mathbf{F}(\acute{X})\mathbf{V}^{-1}\mathbf{H})(\mathbf{H}^T\mathbf{V}^{-1}\mathbf{H})^{-1}(\acute{\mathbf{H}}(\acute{X}) - \mathbf{F}(\acute{X})\mathbf{V}^{-1}\mathbf{H})^T, \end{aligned}$$

with $\hat{\boldsymbol{\beta}} = (\mathbf{H}^T\mathbf{V}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{V}^{-1}\mathbf{y}$. The notation here is as follows:

$$\begin{aligned}
\mathbf{H} &= \mathbf{I}_k \otimes \mathbf{h}(X)^T, \\
\acute{\mathbf{H}}(\acute{X}) &= \mathbf{I}_k \otimes \mathbf{h}(\acute{X})^T, \\
\mathbf{V} &= \mathbf{C}(X, X), \\
\mathbf{F}(\acute{X}) &= \mathbf{C}(\acute{X}, X),
\end{aligned}$$

where, for any designs $A = (\mathbf{a_1}, ..., \mathbf{a_{n_a}})$ and $B = (\mathbf{b_1}, ..., \mathbf{b_{n_b}})$, $\mathbf{h}(A)$ denotes the $q \times 1$ vector whose ith element is $\mathbf{h}(\mathbf{a_i})$ and $\mathbf{C}(A, B)$ denotes the $n_a k \times n_b k$ matrix whose i, jth element is $[\mathbf{C}(\mathbf{a_s}, \mathbf{b_t})]_{uv}$ with $s = i \pmod{n}_a$, $t = j \pmod{n}_b$, $u = \lceil i/n_a \rceil$, and $v = \lceil j/n_b \rceil$.

The conditioning on the covariance function hyperparameters $\boldsymbol{\theta}$ cannot, in general, be removed analytically. A fully Bayesian approach is to update $\boldsymbol{\theta}$ using MCMC, but then we no longer have a closed form for the posterior distribution, and predictions must be presented as a sample. This approach increases the computational burden, since each MCMC update requires the inversion of the full $nk \times nk$ matrix $\mathbf{V}$, potentially making the emulator itself slow to use. Following Kennedy and O'Hagan (2001), Conti and O'Hagan (2010), Rougier et al. (2009), and others, we estimate $\boldsymbol{\theta}$ and treat it as known. As this does not take into account the uncertainty in $\boldsymbol{\theta}$, we use detailed diagnostics to validate the resulting predictive distribution.

2.1 Modeling choices

Given the simulator, we must choose suitable mean and covariance functions $\mathbf{m}(.)$ and $\mathbf{C}(.,.)$. As the simulator is not a realisation of a Gaussian process, we do not suppose that there are unique 'true' functions. Nevertheless, some choices will result in an invalid emulator, in the sense that prediction intervals for simulator outputs at untried inputs will either be consistently too wide or too narrow (i.e. the emulator is consistently 'overconfident' or consistently 'underconfident'; see Bastos and O'Hagan, 2009). Hence we wish to choose

$\mathbf{m}(.)$ and $\mathbf{C}(.,.)$ that minimise posterior uncertainty whilst maintaining a valid emulator.

Prior judgement will be important here, but in practice, we may wish to experiment with different choices, informed by suitable diagnostics (as one would do in ordinary regression analysis). The choices that we explore here, and the main focus of this paper, are nonseparable covariance functions for $\mathbf{C}(.,.)$.

Identifying a suitable $\mathbf{C}(.,.)$ will depend on the choice of $\mathbf{m}(.)$. If we are able to specify a good parametric approximation to $\boldsymbol{\eta}(.)$ using expert opinion (up to the values of some unknown coefficients), for example as in Oakley (2002) and Rougier et al. (2009), then separable covariance functions are likely to be sufficient (or we may even treat the residual process as a white noise process). If the expert is not able to specify a *close* parametric approximation a priori, as will sometimes be the case, then more flexible covariance functions for the residual process may be desirable, as we cannot rely on departures of $\boldsymbol{\eta}(.)$ from $\mathbf{m}(.)$ being small and 'localised'.

In the next section, we review separable covariance functions, as commonly used in emulators, before considering nonseparable alternatives.

2.2 Separable covariance functions

Conti and O'Hagan (2010) define a valid covariance structure using a single function for all of the direct and cross-correlation functions, $c(.,.)_{ij} = c(.,.)\ \forall i, j$. The between-outputs covariance and the spatial correlation are separable:

$$\mathbf{C}(.,.) = \boldsymbol{\Sigma} c(.,.). \tag{3}$$

We refer to the emulator that uses model (1) with the covariance function (3) as *SEP*. A common choice for $c(.,.)$ in GP emulation is the squared-exponential function $c(\mathbf{x}, \mathbf{x}') = \exp\{-\sum_{i=1}^{p} \phi_i (x_i - x_i')^2\}$. This has one 'roughness parameter' ϕ_i corresponding to each input dimension, which controls how rapidly uncertainty in the output grows we move away from a design point in the direction of that input. Rasmussen and Williams (2006) describe

other correlation functions.

An advantage of the separable covariance approach is mathematical tractability. The covariance matrices in the posterior process (2) have the Kronecker product factorizations $\mathbf{V} = \boldsymbol{\Sigma} \otimes \mathbf{A}$, and $\mathbf{F}(\acute{X}) = \boldsymbol{\Sigma} \otimes \mathbf{T}(\acute{X})$, where $\mathbf{A} = c(X, X)$ and $\mathbf{T}(\acute{X}) = c(\acute{X}, X)$, significantly speeding up computation. Also, the separable covariance has a conjugate prior for $\boldsymbol{\Sigma}$: Following Conti and O'Hagan (2010), we use the improper inverse-Wishart type prior $\pi(\boldsymbol{\Sigma}) \propto |\boldsymbol{\Sigma}|^{-\frac{(k+1)}{2}}$, giving a proper inverse-Wishart posterior for $\boldsymbol{\Sigma}$. This allows $\boldsymbol{\Sigma}$ to be analytically integrated out of the posterior process (2), yielding a multivariate Student-t posterior conditional only on $\boldsymbol{\Phi}$:

$$\boldsymbol{\eta}(\acute{X})|\mathbf{y}, \boldsymbol{\Phi} \sim \mathrm{St}_k\{n - q, \mathbf{m}^{\ddagger}(\acute{X}), \hat{\boldsymbol{\Sigma}} \otimes c^{\ddagger}(\acute{X}, \acute{X})\}.$$

where

$$\mathbf{m}^{\ddagger}(\acute{X}) = \acute{\mathbf{H}}(\acute{X})\hat{\boldsymbol{\beta}} + \mathbf{T}(\acute{X})\mathbf{A}^{-1}(\mathbf{y} - \mathbf{H}\hat{\boldsymbol{\beta}}), \tag{4}$$

$$\begin{aligned} c(\acute{X}, \acute{X}) = c(\acute{X}, \acute{X}) &- \mathbf{T}(\acute{X})\mathbf{A}^{-1}\mathbf{T}(\acute{X})^T \\ &+ (\mathbf{h}(\acute{X})^T - \mathbf{T}(\acute{X})\mathbf{A}^{-1}\mathbf{h}(X)^T)(\mathbf{h}(X)\mathbf{A}^{-1}\mathbf{h}(X)^T)^{-1} \\ &\times (\mathbf{h}(\acute{X})^T - \mathbf{T}(\acute{X})\mathbf{A}^{-1}\mathbf{h}(X)^T)^T, \end{aligned} \tag{5}$$

and $\hat{\boldsymbol{\Sigma}} = (n - q - k - 1)^{-1}(\mathbf{Y} - \mathbf{H}\hat{\mathbf{B}})^T\mathbf{A}^{-1}(\mathbf{Y} - \mathbf{H}\hat{\mathbf{B}})$, with $\hat{\mathbf{B}}$ the $q \times k$ matrix such that $\mathrm{vec}(\hat{\mathbf{B}}) = \hat{\boldsymbol{\beta}}$, and $\mathbf{Y}$ the $n \times k$ matrix such that $\mathrm{vec}(\mathbf{Y}) = \mathbf{y}$. This posterior is still conditional on roughness parameters $\boldsymbol{\Phi}$, which cannot be analytically integrated out.

This residual covariance has been used in a number of applications (Bhattacharya, 2007, Kennedy et al., 2008, Rougier et al., 2009). However, it has some restrictive implications. First, there is only one spatial correlation function in the model, applicable to every output. That says that we judge the residual process of every output to have the same correlation length. Second, it can be shown (O'Hagan, 1998) that separability in the covariance is equivalent to a kind of Markov property. If we partition the outputs as $\boldsymbol{\eta}(\mathbf{x})^T = [\boldsymbol{\eta}_1(\mathbf{x})^T, \boldsymbol{\eta}_2(\mathbf{x})^T]$,

then, given regression coefficients $\boldsymbol{\beta}$,

$$\text{cov}[\boldsymbol{\eta}_1(\mathbf{x}'), \boldsymbol{\eta}_2(\mathbf{x})|\boldsymbol{\eta}_1(\mathbf{x}), \boldsymbol{\beta}] = \text{cov}[\boldsymbol{\eta}_2(\mathbf{x}'), \boldsymbol{\eta}_1(\mathbf{x})|\boldsymbol{\eta}_2(\mathbf{x}), \boldsymbol{\beta}] = 0, \tag{6}$$

for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. This property has the following interpretation. Suppose that we wish to predict $\boldsymbol{\eta}_1(\mathbf{x}')$ and, for some other point $\mathbf{x}$, we have already observed $\boldsymbol{\eta}_1(\mathbf{x})$. Then observing $\boldsymbol{\eta}_2(\mathbf{x})$ gives us no further information. In other words, there is no sharing of information about the residual across the outputs.

These implications show that separability in the covariance is a strong judgement to make about the simulator residual function. In section 3 we demonstrate examples where this judgement is not suitable, and the limitations of a separable covariance become apparent.

2.3 Nonseparable covariance functions

If the assumption of separability in the covariance function is judged to be too restrictive, then a nonseparable alternative may be sought. A nonseparable covariance function can have a different spatial correlation function for each output. It does not possess the Markov property (6) so, unlike with a separable covariance, there is sharing of information across outputs. In this section we consider two approaches to specifying nonseparable covariance functions: convolution methods, and coregionalization models.

With a nonseparable covariance function there is no conjugate prior for the between-outputs covariance matrix $\boldsymbol{\Sigma}$, so $\boldsymbol{\Sigma}$ cannot be analytically integrated out of the posterior process (2). This means that we must either integrate over $\boldsymbol{\Sigma}$ numerically by sampling from its posterior, or plug an estimate of $\boldsymbol{\Sigma}$ into the posterior and treat it as if it were known.

Convolution methods

A GP can be constructed by taking the convolution of a Gaussian white noise process with an arbitrary smoothing kernel over $\mathcal{X}$. Higdon (2002) notes that a k-output GP can be constructed by, for each output j, choosing a smoothing kernel $\kappa_j(\mathbf{x})$ and convolving it

with a common 'latent' Gaussian white noise process $w(\mathbf{x})$. Using this idea in the model (1), we construct the residual GP $\mathbf{z}(.)$ as

$$z_i(\mathbf{x}) = \int_{\mathcal{X}} \kappa_i(\mathbf{u} - \mathbf{x}) w(\mathbf{u})\, d\mathbf{u} \quad i = 1, ..., k. \tag{7}$$

If $W(\mathbf{x})$ has zero mean and unit variance, then $\mathbf{z}(.)$ has covariance function with elements

$$C_{ij}(\mathbf{x}, \mathbf{x}') = \int_{\mathcal{X}} \kappa_i(\mathbf{u} - \mathbf{x}) \kappa_j(\mathbf{u} - \mathbf{x}')\, d\mathbf{u}. \tag{8}$$

If the smoothing kernels are squared-exponential functions then the spatial correlation function of $\mathbf{z}(.)$ will also be a squared-exponential.

The dependencies between the outputs are introduced by their shared dependence on latent process $W(\mathbf{x})$. Other authors have extended this approach, defining more complex covariance functions by summing multiple convolutions (Boyle and Frean, 2005, Alvarez and Lawrence, 2009). This can allow, for example, outputs to posses both shared and independent features by convolving with a mixture of shared and independent latent processes. Majumdar and Gelfand (2007) define a covariance function for each output directly and take pairwise convolutions to construct the cross-covariance functions. Their resulting model is similar to the convolved latent process models.

A feature of convolution methods is that they focus on constructing (sometimes highly complex) covariance functions for individual outputs, letting the convolution framework automatically define valid cross-covariance functions. There is less emphasis on accurate modeling of the between-output correlations. The covariance defined by equation (8) is fully determined by the parameters in the smoothing kernels and the individual output variances. Since each of the smoothing kernels relates to an individual output, there are no free parameters within the model for directly controlling the between-output correlations. If two outputs have identical smoothing kernels (and so have identical spatial correlation functions), then the correlation between those outputs is forced to be equal to one.

To enable more freedom in the specification of the between-output correlations, we need

more parameters in the model (7). Higdon (2002) partitions the input space $\mathcal{X}$ and restricts the domain over which outputs share their dependence on the underlying white noise process. However, the relationship between the partitioning and the resulting covariance structure is not simple and it is not clear how one should choose the partition in order to define a particular cross-correlation function.

The convolution emulator

We propose an alternative method for introducing additional parameters into the model, using multiple dependent white noise processes. The elements of $\mathbf{z}(.)$ are constructed as

$$z_i(\mathbf{x}) = \int_{\mathcal{X}} \kappa_i(\mathbf{u}-\mathbf{x}) w_i(\mathbf{u})\, d\mathbf{u}, \quad i = 1, ..., k,$$

where $\mathbf{w}(\mathbf{x})^T = (w_1(\mathbf{x}), ..., w_k(\mathbf{x}))$ is a multivariate white noise process with $\mathbb{E}[\mathbf{w}(\mathbf{x})] = \mathbf{0}$ and $\text{cov}[w_i(\mathbf{x}), w_j(\mathbf{x}')] = p_{ij}\delta(\mathbf{x}-\mathbf{x}')$, where p_{ij} are the elements of a correlation matrix $\mathbf{P}$. The covariance function of $\mathbf{z}(.)$ is $\mathbf{C}(.,.)$, with elements

$$C_{ij}(\mathbf{x}, \mathbf{x}') = p_{ij} \int_{\mathcal{X}} \kappa_i(\mathbf{u}-\mathbf{x}) \kappa_j(\mathbf{u}-\mathbf{x}')\, d\mathbf{u}.$$

We use squared-exponential smoothing kernels

$$\kappa_i(\mathbf{x}) = \sigma_i \left[\left(\frac{4}{\pi}\right)^p \prod_{\ell=1}^{p} \phi_i^{(\ell)} \right]^{\frac{1}{4}} \exp\{-2\mathbf{x}^T \mathbf{\Phi}_i \mathbf{x}\} \qquad i = 1, ..., k,$$

where $\mathbf{\Phi}_i$ is the diagonal matrix $\text{diag}(\phi_i^{(1)}, ..., \phi_i^{(p)})$, giving the covariance function

$$C_{ij}(\mathbf{x}, \mathbf{x}') = \tilde{\Sigma}_{ij} \rho_{ij} \exp\{-2(\mathbf{x}-\mathbf{x}')^T \mathbf{\Phi}_i (\mathbf{\Phi}_i + \mathbf{\Phi}_j)^{-1} \mathbf{\Phi}_j (\mathbf{x}-\mathbf{x}')\},$$

where $\tilde{\Sigma}_{ij} = \sigma_i \sigma_j p_{ij}$ and

$$\rho_{ij} = 2^{\frac{p}{2}} \prod_{\ell=1}^{p} \left[(\phi_i^{(\ell)} \phi_j^{(\ell)})^{\frac{1}{4}} (\phi_i^{(\ell)} + \phi_j^{(\ell)})^{-\frac{1}{2}} \right]. \tag{9}$$

Note that $\rho_{ii} = 1$, so the variance function for output i is $C_{ii}(\mathbf{x}, \mathbf{x}') = \sigma_i^2 \exp\{-(\mathbf{x} - \mathbf{x}')^T \mathbf{\Phi}_i (\mathbf{x} - \mathbf{x}')\}$ with roughness parameters $\mathbf{\Phi} = \{\mathbf{\Phi}_1, ..., \mathbf{\Phi}_k\}$. We refer to the emulator that uses this model as *CONV*.

Using a multivariate white noise process in *CONV* we have introduced an additional $\frac{1}{2}k(k-1)$ parameters: the off-diagonal elements of correlation matrix $\mathbf{P}$. For outputs i and j, ρ_{ij} represents the maximum absolute value of the between-outputs correlation allowed by the choice of smoothing kernels and their parameters. The parameter p_{ij} gives the required flexibility by scaling down the maximum value and giving it the appropriate sign. The complete set of covariance hyperparameters is $(\boldsymbol{\sigma^2}, \mathbf{P}, \mathbf{\Phi})$, where $\boldsymbol{\sigma^2} = (\sigma_1^2, ..., \sigma_k^2)^T$ and $\mathbf{\Phi} = \{\mathbf{\Phi}_1, ..., \mathbf{\Phi}_k\}$. For convenience, we parameterize instead by $\boldsymbol{\theta} = (\tilde{\mathbf{\Sigma}}, \mathbf{\Phi})$, with $\tilde{\mathbf{\Sigma}} = \{\tilde{\Sigma_{ij}}\}$ as defined above equation (9).

The linear model of coregionalization

The linear model of coregionalization (LMC) was developed in the field of geostatistics as a tool to model multivariate spatial processes (Journel and Huijbregts, 1978, Wackernagel, 1995, Goulard and Voltz, 1992, Gelfand et al., 2004). The idea behind the LMC is to construct output processes $\mathbf{z}(.)$ as linear combinations of a number of building-block processes. The number of building-block processes is unrestricted, but we use an LMC with k building blocks, written as

$$\mathbf{z}(.) = \mathbf{R}\mathbf{u}(.). \tag{10}$$

The $k \times k$ matrix $\mathbf{R}$ is full-rank, and $\mathbf{u}(.)$ is a vector of k independent zero mean, unit variance GPs with spatial correlation functions $\kappa_1(.,.), ..., \kappa_k(.,.)$ with hyperparameters $\tilde{\Phi}$. The associated covariance function for $\mathbf{z}(.)$ is

$$\begin{aligned} \mathbf{C}(\mathbf{x}, \mathbf{x}') &= \mathbf{R}[\text{diag}\{\kappa_1(.,.), ..., \kappa_k(.,.)\}]\mathbf{R}^T \\ &= \sum_{j=1}^{k} \mathbf{\Sigma_j} \kappa_j(.,.), \end{aligned} \tag{11}$$

where, for $j = 1, ..., k$, $\mathbf{\Sigma_j} = \mathbf{r_j}\mathbf{r_j}^T$, with $\mathbf{r_j}$ the jth column of $\mathbf{R}$.

The covariance function for a single output or a pair of outputs is a weighted sum of the functions $\{\kappa_1(.,.), ..., \kappa_k(.,.)\}$. The weights are determined by the elements of the *coregionalization matrices* $\{\mathbf{\Sigma_j} : j = 1, ..., k\}$. With this covariance function, the covariance matrices in the posterior process (2) become sums of Kronecker products:

$$\mathbf{V} = \sum_{j=1}^{k} \mathbf{\Sigma_j} \otimes \kappa_j(X, X),$$
$$\mathbf{F}(\acute{X}) = \sum_{j=1}^{k} \mathbf{\Sigma_j} \otimes \kappa_j(X, \acute{X}).$$

An advantage of the LMC is that, by composing the overall correlation function as combinations of a set of functions, we can model variation occurring on different scales.

The LMC emulator

We use the LMC for the residual process $\mathbf{z}(.)$ in the emulator prior (1). The between-outputs covariance of the emulator is $\mathbf{\Sigma} = \mathbf{R}\mathbf{R}^T$. Since matrix square-roots are not unique, $\mathbf{\Sigma}$ and $\mathbf{R}$ are not one-to-one, and different choices of $\mathbf{R}$ can lead to multiple models with the same $\mathbf{\Sigma}$. A natural way to simplify the model further is to parameterize by $\mathbf{\Sigma}$ rather than $\mathbf{R}$ and to specify a particular square-root decomposition to obtain $\mathbf{R}$. Gelfand et al. (2004) use the lower-triangular Cholesky decomposition for its computational ease. We argue that, while we may not necessarily gain additional richness through other decompositions (since the number of free parameters to be specified remains the same), other decompositions do produce fundamentally different models. Hence the Cholesky decomposition should not necessarily be used simply because it is computationally convenient.

Under the Cholesky decomposition, $\mathbf{R}$ is lower triangular so the residual process for the ith output is

$$z_i(.) = \sum_{j=1}^{i} r_{ij} u_i(.),$$

where r_{ij} is the i,jth entry of $\mathbf{R}$. We can rewrite the elements of equation (10) as

$$z_i(.) = \sum_{j=1}^{i-1} \alpha_j^{(i)} z_j(.) + r_{ii} u_i(.), \tag{12}$$

for some set of coefficients $\alpha_j^{(i)}$. Thus the model has an ordered form with the ith output residual being constructed as a linear combination of the first $i-1$ output residuals plus an independent Gaussian process. This means that the ordering of the outputs matters, because the ordering should be taken as fixed. A consequence of this structure is the following.

Theorem 2.1 *Let* $\mathbf{z}(.)$, $\mathbf{u}(.)$ *and* $\mathbf{R}$ *be defined as in equation* (10)*, with* $\mathbf{R}$ *lower triangular. Suppose that (a) there exists* $1 < j < k$ *such that the* j*th row of* $\mathbf{R}$ *has non-zero entries to the left of the diagonal, and (b) that the set of correlation functions* $\{\kappa_1(.,.), ..., \kappa_j(.,.)\}$ *is such that for some* $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ *with* $\mathbf{x} \neq \mathbf{x}'$*, and some* $u, v \in \{1, ..., j\}$, $\kappa_u(\mathbf{x}, \mathbf{x}') \neq \kappa_v(\mathbf{x}, \mathbf{x}')$. *Then there exists* $i < j$ *such that*

$$\mathrm{cov}[\eta_i(\mathbf{x}'), \eta_j(\mathbf{x}) | \eta_i(\mathbf{x}), \boldsymbol{\beta}] = 0, \tag{13}$$

$$\mathrm{cov}[\eta_j(\mathbf{x}'), \eta_i(\mathbf{x}) | \eta_j(\mathbf{x}), \boldsymbol{\beta}] \neq 0. \tag{14}$$

(Proof given in the appendix.)

This result has the following interpretation. Let $\mathbf{x}'$ be any point in $\mathcal{X}$. Suppose that, for $i < j$, we wish to predict $\eta_i(\mathbf{x}')$, and, for some other $\mathbf{x} \in \mathcal{X}$, we have observed $\eta_1(\mathbf{x}), ..., \eta_i(\mathbf{x})$. Equation (13) implies that $\eta_i(\mathbf{x}')$ and $\eta_j(\mathbf{x})$ are conditionally independent given $\eta_1(\mathbf{x}), ..., \eta_i(\mathbf{x})$, so observing $\eta_j(\mathbf{x})$ gives no further information. But the argument is asymmetric: under the conditions of the theorem, equation (14) implies that $\eta_j(\mathbf{x}')$ and $\eta_i(\mathbf{x})$ are not conditionally independent given $\eta_j(\mathbf{x})$, so if we wish to predict $\eta_j(\mathbf{x}')$, then $\eta_i(\mathbf{x})$ is informative, even if we have observed $\eta_j(\mathbf{x})$.

If we can establish *a priori* some natural ordering of the outputs then the Cholesky decomposition may be appropriate. For example, Kennedy and O'Hagan (2000) use a

special case of (12) (an autoregressive model) to emulate outputs are that are the results of a computer code run at different levels of sophistication. However, for many simulators there is no obvious hierarchy of dependence in the outputs. It would not be appropriate to impose an arbitrary asymmetry on them, so decompositions of $\mathbf{\Sigma}$ other than the Cholesky should be considered for the LMC.

We propose the eigendecomposition of $\mathbf{\Sigma}$ as a suitable alternative. The eigendecomposition is $\mathbf{R} = \mathbf{\Lambda}\sqrt{\mathbf{D}}\mathbf{\Lambda}^T$, where $\mathbf{\Lambda}$ is the orthogonal matrix of normalized eigenvectors of $\mathbf{\Sigma}$ and $\sqrt{\mathbf{D}}$ is the diagonal matrix of the square roots of the eigenvalues of $\mathbf{\Sigma}$. With this decomposition $\mathbf{R}$ is symmetric and permuting any two rows and the corresponding columns of $\mathbf{\Sigma}$ will have only the effect of permuting the same rows and columns of $\mathbf{R}$. Hence the labeling of the outputs will have no impact on the structure of the model. We refer to the emulator that uses the model (1) with the residual process $\mathbf{z}(.)$ defined by equation (10), with symmetric $\mathbf{R}$, as *LMC*.

Note the difference between the parameterizations of the LMC and the convolution model. Recall that we cannot arbitrarily specify a set of spatial correlation functions and a between-outputs covariance matrix. In *CONV* we have parameters $\boldsymbol{\theta} = (\Phi, \tilde{\mathbf{\Sigma}})$, where the Φ directly control the spatial correlation functions but $\tilde{\mathbf{\Sigma}}$ gives only limited control over the between-outputs covariance matrix. Conversely, in *LMC* we have parameters $\boldsymbol{\theta} = (\tilde{\Phi}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ is the between-outputs covariance matrix, but $\tilde{\Phi}$ gives only limited control over the spatial correlation functions.

3 Case studies

We consider two case studies: a simulator used by engineers to study a mechanical structure, and a simple climate model. We implement the *SEP*, *CONV* and *LMC* emulators, and for comparison, we emulate the various outputs independently using univariate emulators. We refer to this independent outputs approach as *IND*.

We use squared-exponential correlation functions throughout. We estimate the relevant hyperparameters at their maximum likelihood values. Recall that in the case of the separable

covariance, use of the vague conjugate prior allows the between-outputs covariance Σ to be integrated out, giving $p(\boldsymbol{\eta}(.)|\boldsymbol{\Phi}, \mathbf{y})$, the posterior process conditional only on the correlation function hyperparameters. We refer to this as an *integrated emulator*. We can obtain a similar integrated posterior for each of the independent univariate emulators in the *IND* method. This analytic integration is not possible in either of the nonseparable approaches, so instead we estimate all of the covariance function hyperparameters $\boldsymbol{\theta} = (\boldsymbol{\Sigma}, \boldsymbol{\Phi})$ and predict using the posterior $p(\boldsymbol{\eta}(.)|\boldsymbol{\theta}, \mathbf{y})$. We refer to this as a *conditional emulator*. This raises a question: to make a fair comparison of the emulators, should we use integrated or conditional versions of *IND* and *SEP*?

An integrated emulator should, according to theory, give a better representation of posterior beliefs than the conditional posterior since it takes into account the uncertainty in $\boldsymbol{\Sigma}$. But that is only true if the modeling choices we have made (for example the assumption of separability in the covariance) are a good representation of our judgements about the simulator. If our choices are poor then it is possible that the conditional emulator is at least as good as the integrated emulator.

For thoroughness, in both case studies we fit the integrated and the conditional versions of *IND* and *SEP*. However, we find that in all our diagnostics there is little difference between the two versions, and the conclusions we draw about the relative merits of the emulators are the same no matter which version we use. To simplify presentation of the comparisons, we give results from just the conditional versions of *IND* and *SEP*.

In each case study we run the simulator on a Latin hypercube design spanning the input space to obtain n training data. For evaluating the emulators we obtain $\acute{n}$ validation data by running the simulator on a second, independently generated, Latin hypercube design. We use the various emulators to predict the simulator outputs at each member of the validation design and compare the predictions with the true values, using the following diagnostics:

- The root-mean-squared error (RMSE), defined as the square root of the average squared prediction error. We calculate the RMSE for each output individually, so it indicates the output-marginal prediction accuracy of the emulators. To compare

the magnitude of the prediction errors between outputs, we standardize the RMSE for each output by dividing by the range of the corresponding validation data.

- D^{α}, the proportion of $100\alpha\%$ credible intervals that contain the true validation point. A good emulator will have values of D^{α} close to α, so we plot D^{α} against $\alpha \in [0,1]$ and look for deviation from the straight line through the origin with unit slope.

- D_i^{MD}, the squared Mahanalobis distance of validation point i, which, according to (2), should be distributed $\chi^2_{k\acute{n}}$. We compare the observed values of D_i^{MD} to the chi-squared reference distribution; a discrepancy indicates misspecification of the between-outputs covariance structure.

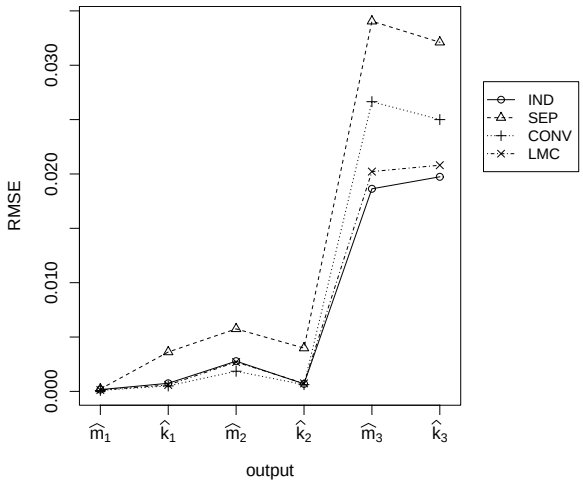
3.1 Case Study 1: A finite element model of an aircraft

The aircraft model of the Garteur benchmark problem was designed by the Garteur Action Group to evaluate ground vibration test techniques (Balmes and Wright, 1997). A finite element modal (FEM) was created as a theoretical representation of the aircraft structure. The FEM has parameters determining the physical properties of the structure, some of which are uncertain. The output of the FEM consists of several pairs of *modal parameters*: a modal mass parameter $\hat{m}_i$ and a modal stiffness parameter $\hat{k}_i$. The subscript i indexes the modal frequencies of vibration of the aircraft. A typical finite element analysis considers a subset of the modal parameters.

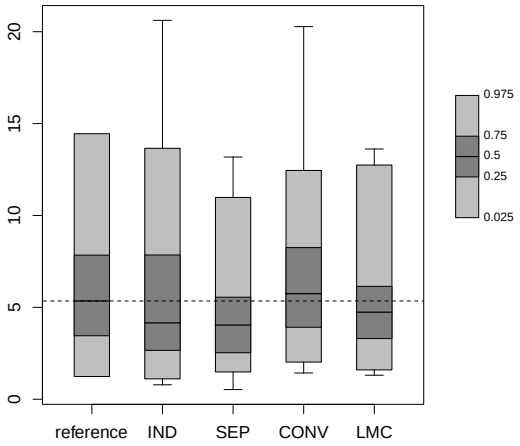
For this case study, we choose $p = 5$ of the uncertain FEM parameters to serve as the inputs to the simulator, which we denote $\mathbf{x}^T = (x_1, ..., x_5)$. The remaining parameters are held fixed at values supplied by the FEM authors. We consider the first three pairs of modal parameters, giving a total of $k = 6$ outputs denoted $\mathbf{y}^T = (y_1, ..., y_6) = (\hat{m}_1, \hat{k}_1, ..., \hat{m}_3, \hat{k}_3)$. We use $n = 50$ runs of the FEM as training data, and a further 50 runs as validation data.

Figure 1a compares the standardized RMSEs of the emulator predictions. *IND* and *LMC* have the best output-marginal prediction accuracy, followed by *CONV* then *SEP*. Box plots of the observed squared Mahanalobis distances D_i^{MD} (Figure 1b) show that the covariances

of the emulators are generally well validated, although *SEP* is a little underconfident.



(a)

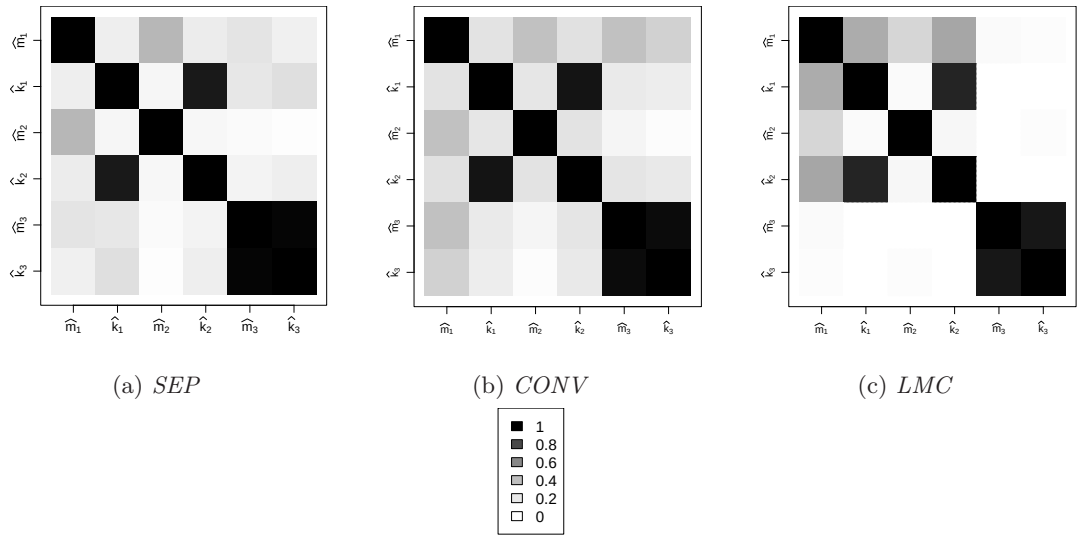


(b)

Figure 1: Case Study 1: (a) Standardized RMSE plotted for each output. (b) Box plots of the observed D^{MD} for each emulator, with a box plot of the chi-squared reference distribution. The legend to the right shows which quantiles are represented in the plots. The whiskers show the range of the observed values. The horizontal dashed line shows the median of the reference distribution.

These diagnostics suggest that the covariance structure of *SEP* is least suitable for this application, and that prediction is better if the outputs have different spatial correlation functions. However, there is little evidence to suggest that the nonseparable covariance structures (in *CONV* and *LMC*) offer a significant advantage over the independent-outputs approach. If we are interested only in marginal predictions of the individual outputs then we should use the *IND* approach. The D_i^{MD} diagnostic suggests that *IND* also gives reasonable joint output predictions. A reason for this is that the estimates of the covariance matrix

$\boldsymbol{\Sigma}$ obtained in the multivariate emulators give that there is mostly low correlation between the outputs, with only two pairs of outputs having a correlation greater than 0.5 (see Figure 2). Ignoring these between-output correlations has little negative impact on the joint distribution of all six outputs.



(a) *SEP* (b) *CONV* (c) *LMC*

Figure 2: Case Study 1: Maps of the estimated between-outputs correlation matrices.

The joint distribution of output predictions may be important when predicting a function that combines correlated outputs. With the FEM, an engineer may be interested in using the outputs to compute the *frequency response function* (FRF) of the mechanical structure (Moens and Vandepitte, 2004). In an undamped structure the FRF as a function of frequency ω is

$$FRF(\omega) = \sum_{i=1}^{n_m} \frac{1}{\hat{k}_i - \omega^2 \hat{m}_i}, \qquad \omega \neq \sqrt{\hat{k}_i/\hat{m}_i} \quad \forall i = 1, ..., n_m. \tag{15}$$

We predict the FRF at a fixed frequency ω^*. The FRF is most heavily influenced by the outputs $\hat{k}_i$ and $\hat{m}_i$ at frequencies that are close to $\sqrt{\hat{k}_i/\hat{m}_i}$. In order to show the effect of correlation between outputs on the FRF, we choose $\omega^* = 200$ Hz, a frequency at which the FRF is heavily influenced by two highly correlated outputs, $\hat{k}_3$ and $\hat{m}_3$ (outputs 5 and 6).

We could, in theory, analytically calculate the predictive distribution of $FRF(\omega^*)$ (as a function of the FEM inputs $\mathbf{x}$) using equation (15) and the joint posterior of the FEM outputs $\mathbf{y}$ from any one of the emulators. This is infeasible, though, due to the non-linearity of (15). Instead, we sample from the joint posterior of $\mathbf{y}$ and substitute the sampled values into (15) to obtain a sample from the posterior of $FRF(\omega^*)$. Since $FRF(\omega^*)$ goes to infinity as ω^* approaches $\sqrt{\hat{k}_i/\hat{m}_i}$ for each $i \in \{1, ..., n_m\}$, we expect the predictive distribution at some validation points to be strongly skewed, so we use the sample median for point predictions.

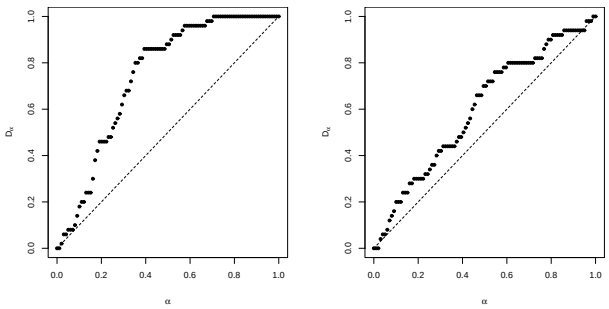
We predict $FRF(\omega^*)$ at the 50 validation points using each of the FEM emulators. The RMSEs (Table 1) show that the relative accuracies of the $FRF(\omega^*)$ predictions broadly correspond to the relative accuracies of the predictions of FEM outputs 5 and 6, with *IND* having the greatest accuracy. However, plots of the D^α diagnostic (Figure 3) show that *IND* has consistently underconfident predictions, caused by the false assumption of independence between the outputs. Note that the reason that the D_i^{MD} diagnostic previously showed *IND* to have reasonable joint output predictions, but here we see a problem with the *IND* predictions of $FRF(\omega^*)$, is that D_i^{MD} gives equal weight to all outputs, whereas $FRF(\omega^*)$ is particularly influenced by two outputs which are highly correlated. The *SEP* emulator has overconfident predictions, due to the misspecification of the spatial correlation functions. For *CONV* and *LMC* the D^α diagnostics are close to the reference line, suggesting that a nonseparable covariance structures is required to adequately model the joint distribution of outputs 5 and 6.

IND	*SEP*	*CONV*	*LMC*
4.24	8.3	13.3	7.97

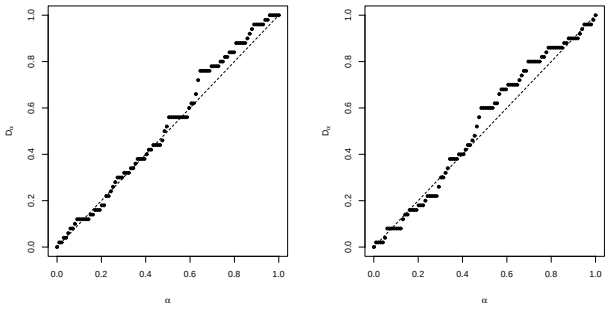
Table 1: Case Study 1: The RMSE for the $FRF(\omega^*)$ validation data. The values have been multiplied by 10^6

3.2 Case Study 2: A simple climate model

Urban and Keller (2010) developed a simple Earth system simulator (referred to here as



(a) *IND* (b) *SEP*



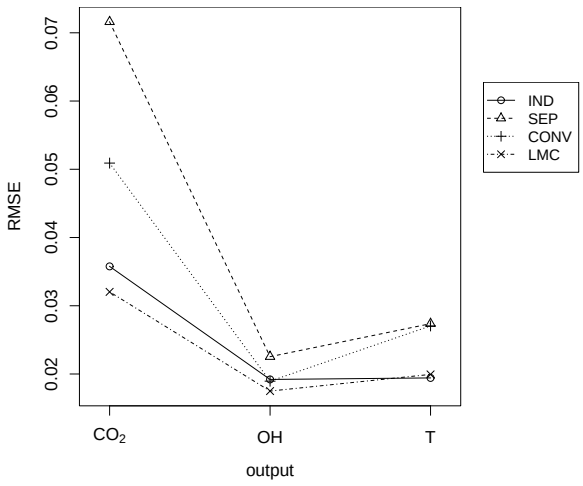
(c) *CONV* (d) *LMC*

Figure 3: Case Study 1: D^{α} plots for predictions of $FRF(\omega^*)$.

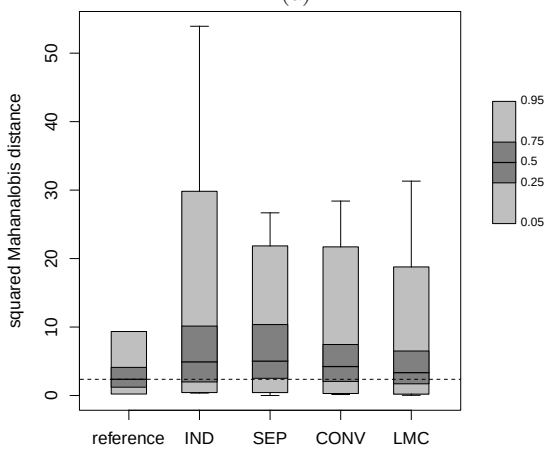
the Simple Climate Model or SCM) to obtain predictions of past and future values of a variety of climate variables. We consider emulating the output values corresponding to year 2000 of three variables: atmospheric CO_2 concentration (CO_2), ocean heat uptake (OH), and global surface temperature anomaly (T). We select five simulator parameters as the variable inputs. We use $n = 55$ runs of the SCM as training data, and we have available a further 90 runs as validation data.

Figure 4a compares the RMSEs of the emulator predictions. The relative output-marginal prediction accuracies of the emulators are similar to those in Case Study 1, with *IND* and *LMC* performing the best, followed by *CONV* then *SEP*. As before, prediction accuracy is better when the outputs have different spatial correlation functions. The box plots of the observed D_i^{MD} in Figure 4b show that all emulators have overconfident joint output predictions, but the problem is worst in *IND*. The reason for this is that there is high correlation between all three outputs, according to the estimates from the multivariate emulators shown in Figure 5.

As in Case Study 1, there is not a strong case for using a nonseparable covariance over



(a)



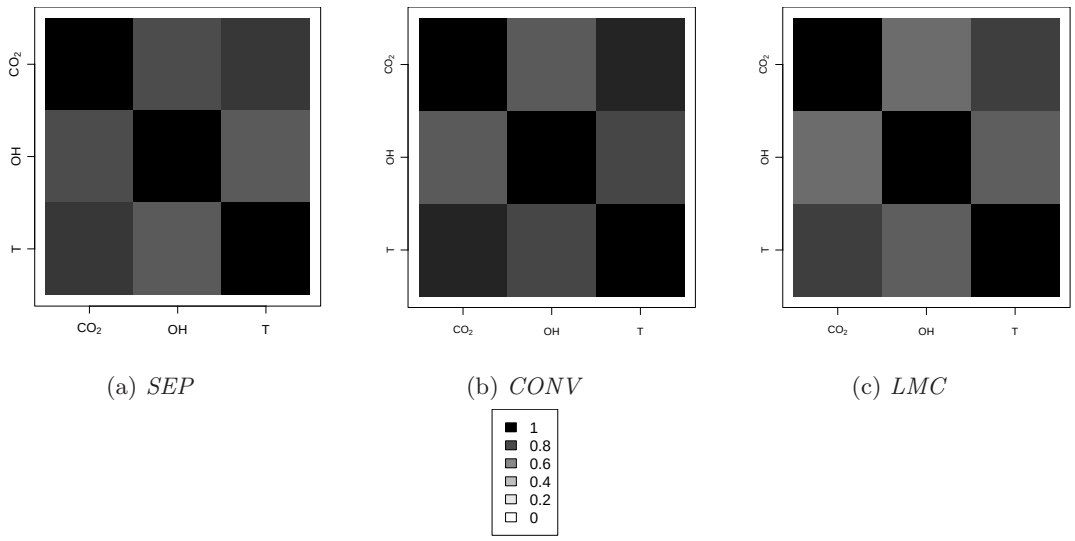
(b)

Figure 4: Case Study 2: (a) Standardized RMSE plotted for each output. (b) Box plots of the observed D^{MD} for each emulator, with a box plot of the chi-squared reference distribution. The legend to the right shows which quantiles are represented in the plots. The whiskers show the range of the observed values. The horizontal dashed line shows the median of the reference distribution.

the independent outputs approach if we are interested only in predicting individual outputs. But again there may be interest in a function of multiple outputs. Jones et al. (2006) model the quantity *gross primary productivity* (GPP) as a function of a number of parameters which include the SCM output variables CO_2 and T:

$$GPP = GPP_{max}\left(\frac{CO_2}{CO_2 - C_{0.5}}\right)\left(1 - \Gamma T_{opt} T + \frac{\Gamma}{2}T^2\right).$$

We consider predicting GPP using fixed values of the additional parameters from the



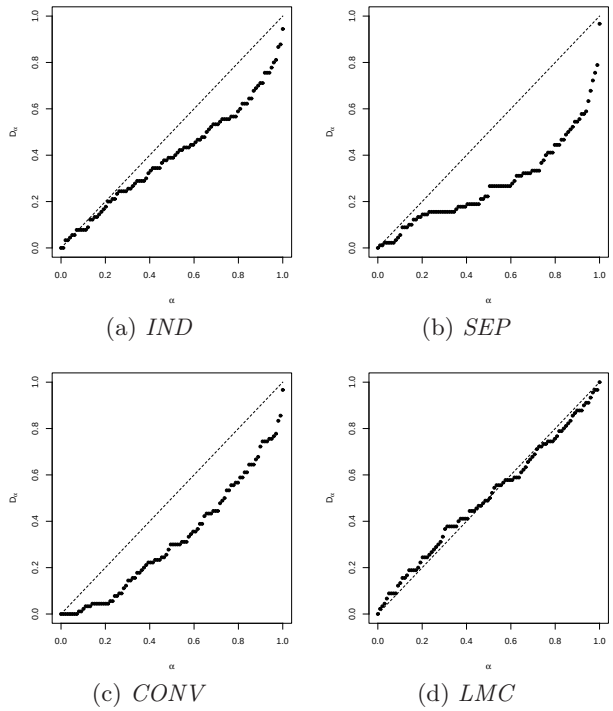
(a) *SEP* (b) *CONV* (c) *LMC*

Figure 5: Case Study 2: Maps of the estimated between-outputs correlation matrices.

ranges suggested by Jones et al. (2006) ($GPP_{max} = 1$, $C_{0.5} = 466$ ppm, $\Gamma = -0.024$ K^{-2} and $T_{opt} = 3.0$ K). We use the same sampling scheme as in Case Study 1 to obtain the predictions, using the sample median for point predictions. As with the raw CO_2 and T outputs, there is little difference between *IND* and *LMC* in terms of prediction accuracy (table 2), and both have greater accuracy than *SEP* and *CONV*. However, the plots of the D^{α} diagnostics in Figure 6 show that the *LMC* posterior has better correspondence to the validation data than that of *IND*. This is because the *IND* method does not account for correlation between CO_2 and T, so predictions are overconfident. Although the correlation is modeled in *SEP* and *CONV*, the inflation in uncertainty that the correlation induces in the GPP predictions is not enough to compensate for the original overconfidence in the raw output predictions from these methods. The *LMC* emulator had the most accurate original predictions of the raw outputs which, combined with its modeling of between-output correlation, lead to it having the best predictive distribution for GPP.

IND	*SEP*	*CONV*	*LMC*
3.30	6.27	4.85	3.00

Table 2: Case Study 2: The RMSE for the GPP validation data. The values have been multiplied by 10^3



(a) *IND* (b) *SEP*

(c) *CONV* (d) *LMC*

Figure 6: Case Study 2: D^α plots for the predictions of GPP.

4 Discussion

In this paper we have developed two methods of constructing emulators with nonseparable residual covariance structures, proposing modifications to convolution and coregionalization methods to make them suitable for use in multivariate emulation. Our convolution-based emulator has additional hyperparameters to control the between-output correlations, and our LMC-based emulator uses the eigendecompostion of the covariance matrix to make it invariant under reordering of the outputs. In both cases, the difference between the nonseparable covariance and a simple separable covariance is substantial in terms of the flexibility offered.

We have compared the covariance structures in two case studies, and found that our nonseparable models offer some advantages. The advantage over the separable covariance

was clear: in both case studies we found that restricting the model to having just one spatial correlation function for all the outputs, as in the separable covariance, resulted in poor predictions of some or all outputs. The advantage over the independent-outputs approach was less clear. We were able to obtain similarly good, or better, marginal predictions of individual outputs from the independent emulators as we were from emulators with either of the nonseparable covariances. However, when predicting scalar-valued functions of multiple outputs we found that ignoring the between-output correlations resulted in either underconfident or overconfident predictions.

A scalar-valued function of multiple outputs could be emulated directly with a univariate emulator, without the interim step of emulating the raw simulator outputs. However, the functions in our case studies contain singularities, which would make direct emulation problematic. Also, the functions contain parameters other than just the simulator output, so a new univariate emulator would have to be built every time these parameters are changed. The advantage of emulating the raw outputs first is that one can the use the raw output emulator to obtain predictions of the function with any configuration of the other parameters.

Comparing the two nonseparable covariances, we found that the LMC approach outperformed the convolution approach in both case studies. We speculate that this may be due to the spatial correlation functions in the models. In the LMC, a consequence of taking linear combinations of a number of independent Gaussian processes is that the spatial correlation function for an individual output is a weighted sum of several basis correlation functions. This allows the modeling of output variation that occurs on several different scales. In the convolution approach we directly assign a single spatial correlation function to each output via the choice of smoothing kernel. We used squared-exponential correlation functions, thus restricting the outputs to each have a single 'range' of correlation. It would be interesting to investigate whether we can improve the convolution approach through use of alternative correlation functions.

We note that, by estimating the covariance hyperparameters in our examples and treat-

ing them as fixed, we do not account for the uncertainty in them. In Case Study 1 there is no evidence of overconfidence in the predictions, so a fully Bayesian treatment of the hyperparameters would not necessarily improve the emulator. In Case Study 2, however, all the emulators are somewhat overconfident, suggesting that the missing uncertainty of the hyperparameters may be significant. A fully Bayesian analysis, using MCMC to integrate over the hyperparameters, could improve things, but to do so is nontrivial. There are a large number of hyperparameters in the model, some of which are matrix valued, and constructing an efficient MCMC algorithm with a reasonable acceptance rate is not easy. While beyond the scope of this paper, we see this as a next step in our research.

We conclude that, if emulating a multiple output simulator, it would be desirable to implement a variety of different emulators (independent, separable and nonseparable), perform diagnostics, and select that which is most fit for purpose. If the outputs have different correlation lengths, and there is interest in joint predictions of multiple outputs, then the nonseparable emulators may be best. However, nonseparable emulators may not be practicable in some very large dimension simulators: we may simply not have enough computing power to invert the full nonseparable covariance matrix. In that case we must choose between the independent and the separable approaches. The former is likely to be best if interest is confined to marginal output prediction, while the latter may be necessary if joint predictions are required.

A Proof of theorem 2.1

In the following proof, for any vectors of variables X and Y of lengths n_X and n_Y respectively, we denote the $n_X \times n_Y$ matrix $\mathrm{cov}[X, Y]$ by $\langle X, Y \rangle$.

Proof: Consider the truncated vector of outputs $[\boldsymbol{\eta}_1(.)^T, \eta_j(.)^T]$, where $\boldsymbol{\eta}_1(.)$ represents the first $j-1$ elements of $\boldsymbol{\eta}(.)$. Truncate and partition $\mathbf{z}(.)$, $\mathbf{u}(.)$ and $\mathbf{R}$ accordingly as

$\mathbf{z}(.)^T = [\mathbf{z}_1(.)^T, z_j(.)^T]$, $\mathbf{u}(.)^T = [\mathbf{u}_1(.)^T, u_j(.)^T]$ and

$$\mathbf{R} = \begin{pmatrix} \mathbf{T}_1 & \mathbf{0} \\ \boldsymbol{\lambda}^T & t_j. \end{pmatrix}.$$

Here, $\mathbf{T}_1$ is a lower triangular $(j-1) \times (j-1)$ matrix, $\boldsymbol{\lambda}$ is a $(j-1) \times 1$ vector with non-zero elements, and t_j is a scalar. Then $\mathbf{z}_1(.) = \mathbf{T}_1 \mathbf{u}_1(.)$ and $z_j(.) = \boldsymbol{\lambda}^T \mathbf{u}_1(.) + t_j u_j(.)$. Given $\mathbf{x} \neq \mathbf{x}' \in \mathcal{X}$, let $\mathbf{C} = \langle \mathbf{u}_1(\mathbf{x}'), \mathbf{u}_1(\mathbf{x}) \rangle$ and $c_j = \langle u_j(\mathbf{x}'), u_j(\mathbf{x}) \rangle$. Since the elements of $\mathbf{u}_1(.)$ are independent processes, $\mathbf{C}$ is diagonal, and by condition (b) in the statement of the theorem the diagonal elements of $\mathbf{C}$ are not all equal.

Proof of equation (13): We have

$$\langle \boldsymbol{\eta}_1(\mathbf{x}'), \eta_j(\mathbf{x}) | \boldsymbol{\eta}_1(\mathbf{x}), \beta \rangle = \langle \mathbf{z}_1(\mathbf{x}'), z_j(\mathbf{x}) | \mathbf{z}_1(\mathbf{x}) \rangle. \tag{16}$$

Form the vector $[\mathbf{z}_1(\mathbf{x}')^T, z_j(\mathbf{x})^T, \mathbf{z}_1(\mathbf{x})^T]$. This has a multivariate normal distribution, so by the usual formula for conditioning in the multivariate normal distribution,

$$\begin{aligned} \langle \mathbf{z}_1(\mathbf{x}'), z_j(\mathbf{x}) | \mathbf{z}_1(\mathbf{x}) \rangle = \langle \mathbf{z}_1(\mathbf{x}'), z_j(\mathbf{x}) \rangle \\ - \langle \mathbf{z}_1(\mathbf{x}'), \mathbf{z}_1(\mathbf{x}) \rangle \langle \mathbf{z}_1(\mathbf{x}), \mathbf{z}_1(\mathbf{x}) \rangle^{-1} \langle \mathbf{z}_1(\mathbf{x}), z_j(\mathbf{x}) \rangle. \end{aligned} \tag{17}$$

Then, noting that $\mathbf{u}_1(\mathbf{x})$ and $u_j(\mathbf{x}')$ are independent for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$,

$$\begin{aligned} \langle \mathbf{z}_1(\mathbf{x}'), z_j(\mathbf{x}) \rangle &= \langle \mathbf{T}_1 \mathbf{u}_1(\mathbf{x}'), \boldsymbol{\lambda}^T \mathbf{u}_1(\mathbf{x}) \rangle = \mathbf{T}_1 \mathbf{C} \boldsymbol{\lambda}, \\ \langle \mathbf{z}_1(\mathbf{x}'), \mathbf{z}_1(\mathbf{x}) \rangle &= \langle \mathbf{T}_1 \mathbf{u}_1(\mathbf{x}'), \mathbf{T}_1 \mathbf{u}_1(\mathbf{x}) \rangle = \mathbf{T}_1 \mathbf{C} \mathbf{T}_1^T, \\ \langle \mathbf{z}_1(\mathbf{x}), \mathbf{z}_1(\mathbf{x}) \rangle &= \langle \mathbf{T}_1 \mathbf{u}_1(\mathbf{x}), \mathbf{T}_1 \mathbf{u}_1(\mathbf{x}) \rangle = \mathbf{T}_1 \mathbf{T}_1^T, \\ \langle \mathbf{z}_1(\mathbf{x}), z_j(\mathbf{x}) \rangle &= \langle \mathbf{T}_1 \mathbf{u}_1(\mathbf{x}), \boldsymbol{\lambda}^T \mathbf{u}_1(\mathbf{x}) \rangle = \mathbf{T}_1 \boldsymbol{\lambda}. \end{aligned}$$

Substituting these expressions into equations (16)-(17), and noting that $\mathbf{T}_1$ is full rank so

it is invertible, we obtain

$$\langle \boldsymbol{\eta}_1(\mathbf{x}'), \eta_j(\mathbf{x}) | \boldsymbol{\eta}_1(\mathbf{x}), \beta \rangle = \mathbf{T}_1 \mathbf{C} \boldsymbol{\lambda} - \mathbf{T}_1 \mathbf{C} \mathbf{T}_1^T (\mathbf{T}_1 \mathbf{T}_1^T)^{-1} \mathbf{T}_1 \boldsymbol{\lambda} = \mathbf{0}.$$

Proof of equation (14): Using similar arguments,

$$\begin{aligned}
\langle \eta_j(\mathbf{x}'), \boldsymbol{\eta}_1(\mathbf{x}) | \eta_j(\mathbf{x}), \beta \rangle &= \langle z_j(\mathbf{x}'), \mathbf{z}_1(\mathbf{x}) \rangle \\
&\quad - \langle z_j(\mathbf{x}'), z_j(\mathbf{x}) \rangle \langle z_j(\mathbf{x}), z_j(\mathbf{x}) \rangle^{-1} \langle z_j(\mathbf{x}), \mathbf{z}_1(\mathbf{x}) \rangle, \\
&= \boldsymbol{\lambda}^T \mathbf{C} \mathbf{T}_1^T - (\boldsymbol{\lambda}^T \mathbf{C} \boldsymbol{\lambda} + t_j^2 c_j)(\boldsymbol{\lambda}^T \boldsymbol{\lambda} + t_j^2)^{-1} \boldsymbol{\lambda}^T \mathbf{T}_1^T \\
&= \left(\boldsymbol{\lambda}^T \mathbf{C} - \frac{\boldsymbol{\lambda}^T \mathbf{C} \boldsymbol{\lambda} + t_j^2 c_j}{\boldsymbol{\lambda}^T \boldsymbol{\lambda} + t_j^2} \boldsymbol{\lambda}^T \right) \mathbf{T}_1^T. \qquad (18)
\end{aligned}$$

We are looking for $i \in \{1, ..., j-1\}$ such that $\langle \eta_j(\mathbf{x}'), \eta_i(\mathbf{x}) | \eta_j(\mathbf{x}), \beta \rangle \neq 0$, so we suppose that

$$\langle \eta_j(\mathbf{x}'), \boldsymbol{\eta}_1(\mathbf{x}) | \eta_j(\mathbf{x}), \beta \rangle = \mathbf{0}, \qquad (19)$$

and seek a contradiction. If equation (19) holds, then equation (18) gives

$$\boldsymbol{\lambda}^T \mathbf{C} = \frac{\boldsymbol{\lambda}^T \mathbf{C} \boldsymbol{\lambda} + t_j^2 c_j}{\boldsymbol{\lambda}^T \boldsymbol{\lambda} + t_j^2} \boldsymbol{\lambda}^T, \qquad (20)$$

(using the fact that $\mathbf{T}_1$ is full rank). Since $\boldsymbol{\lambda}$ has no zero elements, equation (20) implies that $\mathbf{C}$ is a scalar multiple of the identity matrix, contradicting the fact that the diagonal elements of $\mathbf{C}$ are not all equal. ∎

References

Alvarez, M. and Lawrence, N. D. (2009). Sparse convolved gaussian processes for multi-output regression, in *Advances in Neural Information Processing Systems 21*, edited by Koller, D., Schuurmans, D., Bengio, Y. and Bottou, L., pp. 57–64, MIT Press.

Balmes, E. and Wright, J. (1997). GARTEUR group on ground vibration testing: results from the test of a single structure by 12 laboratories in Europe, in *Proceedings of the International Modal Analysis Conference, Orlando*.

Bastos, L. S. and O'Hagan, A. (2009). Diagnostics for gaussian process emulators, *Technometrics*, **51 (4)**: 425–438.

Bayarri, M., Berger, J., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R., Paulo, R., Sacks, J. and Walsh, D. (2007). Computer model validation with functional output, *Annals of Statistics*, **35 (5)**: 1874–1906.

Bayarri, M. J., Berger, J. and Steinberg, D. M. (2009a). Special issue on computer modeling, *Technometrics*, **51 (4)**: 353–353.

Bayarri, M. J., Berger, J. O., Kennedy, M. C., Kottas, A., Paulo, R., Sacks, J., Cafeo, J. A., Lin, C.-H. and Tu, J. (2009b). Predicting vehicle crashworthiness: Validation of computer models for functional and hierarchical data, *Journal of the American Statistical Association*, **104 (487)**: 929–943.

Bhattacharya, S. (2007). A simulation approach to Bayesian emulation of complex dynamic computer models, *Bayesian Analysis*, **2**: 783–816.

Boyle, P. and Frean, M. (2005). Dependent gaussian processes, in *Advances in Neural Information Processing Systems 17*, edited by Saul, L. K., Weiss, Y. and Bottou, L., pp. 217–224, MIT Press.

Conti, S. and O'Hagan, A. (2010). Bayesian emulation of complex multi-output and dynamic computer models, *Journal of statistical planning and inference*, **140 (3)**: 640–651.

Currin, C., Mitchell, T., Morris, M. and Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments, *Journal of the American Statistical Association*, **86**: 953–963.

Gelfand, A. E., Schmidt, A. M., Banerjee, S. and Sirmans, C. F. (2004). Nonstationary multivariate process modelling through spatially varying coregionalization, *Test*, **13 (2)**: 1–50.

Goldstein, M. and Rougier, J. (2006). Bayes linear calibrated prediction for complex systems, *Journal of the American Statistical Association*, **101 (475)**: 1132–1143.

Goulard, M. and Voltz, M. (1992). Linear coregionalization model: Tools for estimation and choice of cross-variogram matrix, *Journal Mathematical Geology*, **21 (3)**: 269–286.

Higdon, D. (2002). Space and space-time modeling using process convolutions, in *Quantitative Methods for Current Environmental Issues*, edited by Anderson, C. W., Barnett, V., Chatwin, P. C. and El-Shaarawi, A., Springer.

Higdon, D., Gattiker, J., Williams, B. and Rightley, M. (2008). Computer model calibration using high-dimensional output, *Journal of the American Statistical Association*, **103**: 570–583.

Johnson, M., Moore, L. and Ylvisaker, D. (1990). Minimax and maximin distance designs, *Journal of Statistical Planning and Inference*, **26**: 131–148.

Jones, C., Cox, P. and Huntingford, C. (2006). Climate-carbon cycle feedbacks under stabilization: uncertainty and observational constraints, *Tellus*, **58 (5)**: 603–613.

Journel, A. G. and Huijbregts, C. J. (1978). *Mining Geostatistics*, Academic Press.

Kennedy, M., Anderson, C., O'Hagan, A., Lomas, M., Woodward, I., Gosling, J. and Heinemeyer, A. (2008). Quantifying uncertainty in the biospheric carbon flux for England and Wales, *Journal of the Royal Statistical Society: Series A(Statistics in Society)*, **171 (1)**: 109–135.

Kennedy, M. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available., *Biometrika*, **87**: 1–13.

Kennedy, M. and O'Hagan, A. (2001). Bayesian calibration of computer models (with discussion), *Journal of the Royal Statistical Society. Series B*, **63**: 425–464.

Majumdar, A. and Gelfand, A. (2007). Spatial modeling for multivariate environmental data using convolved covariance functions, *Math. Geology*, **39 (2)**: 225–245.

McFarland, J., Mahadevan, S., Romero, V. and Swiler, L. (2008). Calibration and Uncertainty Analysis for Computer Simulations with Multivariate Output, *AIAA journal*, **46 (5)**: 1253.

McKay, M. D., Beckman, R. J. and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics*, **21 (2)**: 239–245.

Moens, D. and Vandepitte, D. (2004). An interval finite element approach for the calculation of envelope frequency response functions, *International Journal for Numerical Methods in Engineering*, **61 (14)**: 2480–2507.

Oakley, J. (2002). Eliciting gaussian process priors for complex computer codes, *Journal of the Royal Statistical Society. Series D (The Statistician)*, **51 (1)**: 81–97.

Oakley, J. E. and O'Hagan, A. (2004). Probabilistic sensitivity analysis of complex models: a Bayesian approach, *Journal of the Royal Statistical Society. Series B*, **66 (3)**: 751–769.

O'Hagan, A. (1998). A markov property for covariance structures, Nottingham University Statistics Research Report 98-13.

Qian, P. Z. G., Wu, H. and Wu, C. F. J. (2008). Gaussian process models for computer experiments with qualitative and quantitative factors, *Technometrics*, **50 (3)**: 383–396.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*, MIT Press.

Rougier, J. (2008). Efficient emulators for multivariate deterministic functions, *Journal of Computational and Graphical Statistics*, **17 (4)**: 827–843.

Rougier, J., Guillas, S., Maute, A. and Richmond, A. (2009). Expert knowledge and multivariate emulation:the thermosphere-ionosphere electrodynamics general circulation model (tie-gcm), *Technometrics*, **51 (4)**: 414–424.

Sacks, J., Welch, W., Mitchell, T. and Wynn, H. (1989). Design and analysis of computer experiments, *Statistical Science*, **4**: 409–435.

Saltelli, A. and Tarantola, S. (2002). On the relative importance of input factors in mathematical models: safety assessment for nuclear waste disposal, *Journal of the American Statistical Association*, **97 (459)**: 702–709.

Santner, T. J., Williams, B. and W, N. (2003). *The Design and Analysis of Computer Experiments*, Springer-Verlag.

Urban, N. M. and Keller, K. (2010). Probabilistic hindcasts and projections of the coupled climate, carbon cycle, and atlantic meridional overturning circulation systems: A bayesian fusion of century-scale observations with a simple model, *Tellus A*, in revision.

Wackernagel, H. (1995). *Multivariate Geostatistics*, Springer.